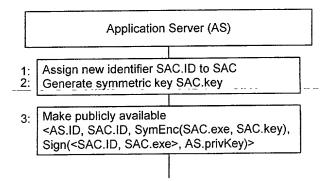


Fig. 1: Application Framework

Fig. 2: SAC Self-Publishing



Application Server (AS) Trust Server (TS) Coprocessor (Cp) Generate one-time key pair (pubKey, privKey) 2: pubKey Assign new certID Performed inside HSM & with atomicity: Compute Sign(<certID, pubKey>, TS.privKey); Record <certiD, pubKey, Cp.ID> Sign(<certID, pubKey>, TS.privKey) SAC.ID, certID, pubKey Sign(<certID, pubKey>, TS.privKey) Verify TS signature; Generate SAC individualization data "blob" & non-secret identifying info for "blob", "blobTag", Record <certID, blob, blobTag> Enc(<blob, blobTag, SAC.key>, pubKey), 10: Sign(Enc(<blob, blobTag, SAC.key>, pubKey), AS.privKey) Verify AS signature; Decrypt message 12: certID, AS.ID, SAC.ID, H(blob) Performed inside HSM & with atomicity: Compute Sign(<certID, AS.ID, H(blob)>, TS.privKey); Verify that certID has not been assigned Record <certID, AS.ID, SAC.ID, H(blob)> 16: Sign(<certID, AS.ID, H(blob)>, TS.privKey) 17: blobTag, Sign(<certID, AS.ID, H(blob)>, TS.privKey) Verify TS signature Mark blob as activated

Fig. 3: Coupon Collection & Redemption

Application Server (AS) Trust Server (TS) Assign new SAC.number; Record SAC.number 3: SAC.number Note: SAC.number is part of SAC.ID. Generate SAC-series symmetric key AS.key; Generate SAC-series tracking secret AS.track; SAC.ID=<SAC.number, SAC.version> Record <SAC.number, AS.key, AS.track> SAC.number, Enc(<AS.track, AS.key, SAC.number>, TS.pubKey) Performed inside HSM & with atomicity: Generate SAC-series symmetric key SAC.key; Compute SAC.assign = Enc(<TS.local, SAC.number, AS.track, AS.key, SAC.key>, Note: TS.local is a TS.pubKey) secret secured by TS HSM Verify that SAC.number has not been assigned 10: Record <SAC.number, SAC.assign>

Fig. 4: SAC-Series Initialization

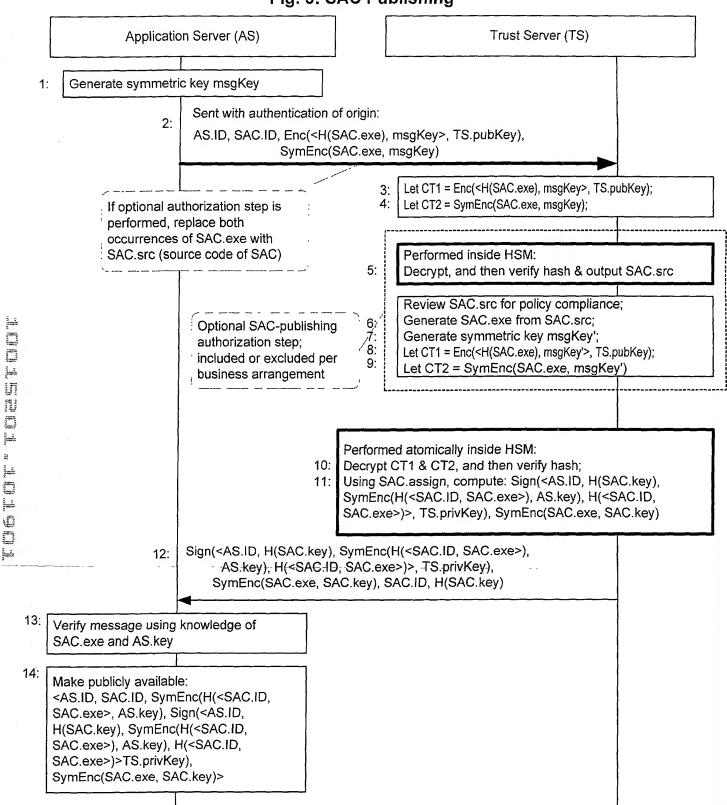


Fig. 6: SAC-Series Bulk Individualization

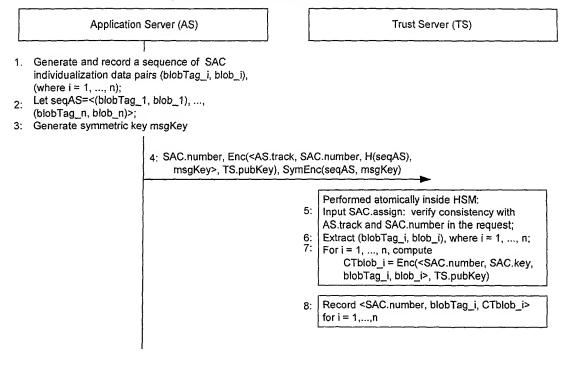


Fig. 7: SAC Permissioning (into Coprocessor): Installation and Individualization

